WE CLAIM

1. A data processor core comprising:

a memory access interface portion operable to perform data transfer operations between an external data source and at least one memory associated with said data processor core;

a data processing portion operable to perform data processing operations;

a read/write port operable to transfer data from said processor core to at least two buses, said at least two buses being operable to provide data communication between said processor core and said at least one memory, said at least one memory comprising at least two portions, each of said at least two buses being operable to provide data access to respective ones of said at least two portions;

arbitration logic associated with said read/write port; wherein

said arbitration logic is operable to route a data access request requesting access of data in one portion of said at least one memory received from said memory access interface to one of said at least two buses providing access to said one portion of said at least one memory and to route a further data access request requesting access of data in a further portion of said at least one memory received from said data processing portion to a further one of said at least two buses providing access to said further portion of said at least one memory, said routing of said data access requests being performed during the same clock cycle.

2. A data processor core according to claim 1, said arbitration logic being operable to select one of said at least two buses to route said data access request to, in

dependence upon an address location within said at least one memory associated with said data access request.

3. A data processor core according to claim 2, wherein said at least two portions of said memory comprise an instruction portion operable to store instructions and at least one data portion operable to store data items, said arbitration logic being operable to route said data access request to a first one of said at least two buses providing access to said instruction portion when data to be transferred is an instruction and to route said data access request to a second of said at least two buses providing access to said at least one data portion when data to be transferred is a data item.

4. A data processor core according to claim 3, wherein said at least one data portion comprises two data portions, an even data portion operable to store data having an even address and an odd data portion operable to store data having an odd address, said read/write port being operable to transfer data between said processor core and said at least one memory via three buses, a first bus providing access to said instruction portion, a second bus providing access to said odd data portion and a third bus providing access to said even data portion, and said arbitration logic being operable to route a data access request to said first bus when data to be transferred is an instruction, to said second bus when data to be transferred is a data item associated with an odd address and to said third bus when data to be transferred is a data item associated with an even address.

5. A data processor core according to claim 1, wherein said arbitration logic is operable in response to receipt of a data access request from said memory access interface portion and a data access request from said data processing portion, both data access requests requesting access to data in one portion of said at least one memory, to route said data access request from said memory access interface portion to one of said at least two buses providing data access to said one portion of said at least one memory before routing said request from said processing portion to said one of said at least two buses.

6. A data processor core according to claim 1, said arbitration logic being operable to detect a wait request from at least one busy portion of said at least one memory, said arbitration logic being operable not to route any data access requests to said busy portion until said wait request is no longer detected.

7. A data processing apparatus comprising:

a data processor core according to claim 1; and

at least one memory, said at least one memory being divided into at least two portions; and

at least two buses, each bus allowing data access to a respective portion of said at least two portions of said at least one memory.

8. A data processing apparatus according to claim 7, wherein said at least one memory is divided into three portions, an instruction portion operable to store instructions, and two data portions, an even data portion operable to store data having

an even address and an odd data portion operable to store data having an odd address, said data processing apparatus comprising three buses said read/write port being operable to transfer data between said processor core and said at least one memory via said three buses, a first bus providing access to said instruction portion, a second bus providing access to said odd data portion and a third bus providing access to said even data portion.

9. A data processing apparatus according to claim 7, wherein said at least one memory is a tightly coupled memory.

10. A method of transferring data between an external data source and at least one memory associated with a data processor core, said data processor core comprising a memory access interface portion operable to perform data transfer operations between said external data source and said at least one memory associated with said data processor core and a data processing portion operable to perform data processing operations, said method comprising the steps of:

in response to a data access request requesting access of data in one portion of said at least one memory received from said memory access interface portion and a data access request requesting access to data in a further portion of said at least one memory received from said data processing portion, routing said data access request received from said memory access interface portion to one of at least two buses, said one of said at least two buses providing access to said one portion of said at least one memory, and routing said data access request received from said data processing portion to a further of said at least two buses, said further bus providing access to said

further portion of said at least one memory, said routing of said data access requests being performed during the same clock cycle.

11. A method according to claim 10, wherein said step of routing data access requests to respective data buses is done in dependence upon an address location within said at least one memory associated with said data access request.

12. A method according to claim 10, wherein said at least two portions of said memory comprise an instruction portion operable to store instructions and at least one data portion operable to store data items, said step of routing said data access requests being operable to route a data access request to one of said at least two buses providing access to said instruction portion when data to be transferred is an instruction and to route said data access request to another of said at least two buses providing access to said at least one data portion when data to be transferred is a data item.

13. A method according to claim 12, wherein said at least one data portion comprises two data portions, an even data portion operable to store data having an even address and an odd data portion operable to store data having an odd address, said routing step being operable to route data accesses to one of three buses, a first bus providing access to said instruction portion, a second bus providing access to said odd data portion and a third bus providing access to said even data portion, and said routing step being operable to route a data access request to said first bus when data to be transferred is an instruction, to said second bus when data to be transferred is a data

item associated with an odd address and to said third bus when data to be transferred is a data item associated with an even address.

14. A method according to claim 10, wherein said routing step in response to receipt of a data access request from said memory access interface portion and a data access request from said data processing portion, both data access requests requesting access to data in a portion of said at least one memory accessed by one of said at least two buses, routes said data access request from said memory access interface portion to said one of said at least two buses before routing said request from said processing portion to said one of said at least two buses.

15. A method according to claim 10, said routing step detecting a wait request from at least one busy portion of said at least one memory, and in response to detection of said wait request not routing any data access requests to said busy portion until said wait request is no longer detected.

16. A method according to claim 10, wherein said at least one memory is divided into three portions, an instruction portion operable to store instructions, and two data portions, an even data portion operable to store data having an even address and an odd data portion operable to store data having an odd address, said routing step routing a received data access request to one of three buses, a first bus providing access to said instruction portion, a second bus providing access to said odd data portion and a third bus providing access to said even data portion, in dependence upon an address of said data associated with said data access request.

17. Arbitration logic operable to control a data processor to perform the steps of the method according to any one of claims 10 to 16.